

# AI Assignment: Basic List Processing

## Abstract:

This assignment is a basic introduction to Common Lisp list structure. The first two assignments are a reproduction of some of the professor's lessons. The final task we use the information we gained from the prior tasks to follow instructions in creating and manipulating a few more lists.

## Task 1: Mimic "Lisp Session: CAR, CDR and CONS"

```
[1]> ( CAR '( BLUE RED YELLOW ) )
BLUE
[2]> ( CDR '( BLUE RED YELLOW ) )
(RED YELLOW)
[3]> ( CAR '( ( 1 2 ) BUCKLE MY SHOE ) )
(1 2)
[4]> ( CDR '( ( 1 2 ) BUCKLE MY SHOE ) )
(BUCKLE MY SHOE)
[5]> ( CAR ("SUNSHINE")
)

*** - EVAL: "SUNSHINE" is not a function name; try using a symbol instead
The following restarts are available:
USE-VALUE      :R1      Input a value to be used instead.
ABORT          :R2      Abort main loop
Break 1 [6]> :A
[7]> ( CAR '( "SUNSHINE" ) )
"SUNSHINE"
[8]> ( CDR '( "SUNSHINE" ) )
NIL
[9]> ( CONS 'ESPRESSO '(LATTE CAPPUCCINO) )
(ESPRESSO LATTE CAPPUCCINO)
[10]> ( CONS '(A B C) '(1 2 3) )
((A B C) 1 2 3)
[11]> ( CONS 'SYMBOL ( ) )
(SYMBOL)
```

Task 2: Mimic “Redacted Lisp Session: Three additional referencers and constructors”

```
[1]> ( setf oo-languages '(simula smalltalk java clos) )
(SIMULA SMALLTALK JAVA CLOS)
[2]> oo-languages
(SIMULA SMALLTALK JAVA CLOS)
[3]> 'oo-languages
OO-LANGUAGES
[4]> (quote oo-languages)
OO-LANGUAGES
[5]> (car oo-languages)
SIMULA
[6]> (cdr oo-languages)
(SMALLTALK JAVA CLOS)
[7]> (car (cdr oo-languages))
SMALLTALK
[8]> (cdr (cdr oo-languages))
(JAVA CLOS)
[9]> (cadr oo-languages)
SMALLTALK
[10]> (cddr oo-languages)
(JAVA CLOS)
[11]> (first oo-languages
)
SIMULA
[12]> (second oo-languages)
SMALLTALK
[13]> (third oo-languages
)
JAVA
[14]> (nth 2 oo-languages)
JAVA
[15]> (setf numbers '(1 2 3))
(1 2 3)
```

## Task 2: Mimic “Redacted Lisp Session: Three additional referencers and constructors” Cont...

```
[16]> (setf letters '(a b c))
(A B C)
[17]> (cons numbers letters)
((1 2 3) A B C)
[18]> (list numbers letters)
((1 2 3) (A B C))
[19]> (append numbers letters)
(1 2 3 A B C)
[20]> (list numbers (cdr numbers) (cddr numbers))
((1 2 3) (2 3) (3))
[21]> (append numbers (cdr numbers) (cddr numbers))
(1 2 3 2 3 3)
[22]> (seft elle '(ant bat cat dog eel) )

*** - EVAL: undefined function SEFT
The following restarts are available:
USE-VALUE      :R1      Input a value to be used instead of (FDEFINITION 'SEFT).
RETRY          :R2      Retry
STORE-VALUE    :R3      Input a new value for (FDEFINITION 'SEFT).
ABORT          :R4      Abort main loop
Break 1 [23]> :a
[24]> (seft elle '(ant bat cat dog eel) )
(ANT BAT CAT DOG EEL)
[25]> (car (cdr (cdr (cdr elle) ) ) )
DOG
[26]> (nth 3 elle )
DOG
```

```
[27]> (setf a 'apple b 'peach c 'cherry )
CHERRY
[28]> (cons a (cons b (cons c ())))
(APPLE PEACH CHERRY)
[29]> (list a b c )
(APPLE PEACH CHERRY)
[30]> (setf x '(red blue) y '(green yellow) )
(GREEN YELLOW)
[31]> (cons (car x) (cons (car (cdr x)) y))
(RED BLUE GREEN YELLOW)
[32]> (append x y )
(RED BLUE GREEN YELLOW)
```

### Task 3: Create a Lisp session according to specification

```
[1]> ( SETF ENGLISH '(ONE TWO THREE FOUR))
(ONE TWO THREE FOUR)
[2]> ( SETF FRENCH '(UN DEUX TROIS QUATRE))
(UN DEUX TROIS QUATRE)
[3]> ( SETF PAIR1 (LIST (CAR ENGLISH) (CAR FRENCH) ) )
(ONE UN)
[4]> ( SETF PAIR2 (LIST (CAR (CDR ENGLISH) ) (CAR (CDR FRENCH) ) ) )
(TWO DEUX)
[5]> ( SETF PAIR3 (LIST (NTH 2 ENGLISH) (NTH 2 FRENCH) ) )
(THREE TROIS)
[6]> ( SETF PAIR4 (LIST (NTH 3 ENGLISH) (NTH 3 FRENCH) ) )
(FOUR QUATRE)
[7]> ( SETF DICTIONARY (LIST PAIR1 PAIR2 PAIR3 PAIR4) )
((ONE UN) (TWO DEUX) (THREE TROIS) (FOUR QUATRE))
[8]> ( SETF EF-WORDS (APPEND PAIR1 PAIR2 PAIR3 PAIR4) )
(ONE UN TWO DEUX THREE TROIS FOUR QUATRE)
[9]> ( SETF ALT-WORDS (APPEND ENGLISH FRENCH) )
(ONE TWO THREE FOUR UN DEUX TROIS QUATRE)
```